5. Mining Rare and Frequent Events in Multi-camera Surveillance Video

Valery A. Petrushin

Summary. This chapter describes a method for unsupervised classification of events in multicamera indoors surveillance video. This research is a part of the Multiple Sensor Indoor Surveillance (MSIS) project, which uses 32 webcams that observe an office environment. The research was inspired by the following practical problem: how automatically classify and visualize a 24-h long video captured by 32 cameras? The self-organizing map (SOM) approach is applied to event data for clustering and visualization. One-level and two-level SOM clustering are used. A tool for browsing results allows exploring units of the SOM maps at different levels of hierarchy, clusters of units, and distances between units in 3D space. A special technique has been developed to visualize rare events.

5.1 Introduction

The rapidly increasing number of video cameras in public places and business facilities, such as airports, streets, highways, parking lots, shopping malls, hospitals, hotels, and governmental buildings can create many opportunities for public safety and business applications. Such applications range from surveillance for threat detection in airports, schools and shopping malls, monitoring highways, parking lots and streets, to customer tracking in a bank or in a store for improving product displays and preventing thefts, to detecting unusual events in a hospital, and monitoring elderly people at home, etc. These applications require the ability automatically detecting and classifying events by analyzing video or imagery data.

In spite of that video surveillance has been in use for decades, the development of systems that can automatically detect and classify events is an active research area. Many papers have been published in recent years. In most of them specific classifiers are developed that allow recognizing objects such as people and vehicles and tracking them [1–3] or recognizing relationships between objects (e.g., a person standing at an ATM machine) and actions (e.g., a person picks up a cup) [4]. The others propose general approaches for event identification using clustering. In [5] the authors segment raw surveillance video into sequences of frames that have motion, count the proportion of foreground pixels for segments of various lengths, and use a multilevel hierarchical clustering to group the segments. The authors also propose a

measure of abnormality for a segment that is a relative difference between average distance for elements of the cluster and average distance from the sequence to its nearest neighbors. The weaknesses of the approach are as follows.

- Segments of higher motion often are subsequences of segments of lower average motion and when they are clustered the subsequences of the same event belong to different clusters.
- Location and direction of movement of the objects are not taken into account.
- The other features, such as color, texture, and shape, which could be useful for distinguishing events, are not taken into account.

The authors of [6] describe an approach that uses a 2D foreground pixels' histogram and color histogram as features for each frame. The features are mapped into 500 feature prototypes using the vector quantization technique. A surveillance video is represented by a number of short (4 s) overlapping video segments. The relationship among video segments and their features and among features themselves is represented by a graph in which edges connect the segments to features and features to features. The weights on the edges reflect how many times each feature occurred in each video segment, and similarity among features. To visualize the graph, it is embedded in a 3D space using the spectral graph method. To categorize the video segments, the authors use the k-means clustering on the video segments' projections. The larger clusters are defined as usual events, but small and more isolated clusters as unusual events. A new video segment can be classified by embedding it into common space and applying k-nearest neighbor classifier. The advantages of this approach are the following:

- Taking into account similarity among features.
- Attractive visualization of results.

But the disadvantages are

- High computational complexity of the graph embedding method.
- Dependence of the results on the length of video segments.

Developing techniques for visualization of large amount of data always attracted attention of data mining researchers. Visualization is an indispensable part of the data exploration process. Finding efficient algorithms for event detection and summarization, skimming and browsing large video and audio databases are the major topics of multimedia data mining [7, 8]. Many visualization techniques have been developed in traditional data mining. They use clustering that follows by a mapping into 2D or 3D space. For example, using the principal component analysis the data mapped into principal component space and then visualized using two or three first components. Another well-known and widely used approach is Self-Organizing Maps (SOM) or Kohonen Neural Networks [9]. This approach has been applied for analysis and visualization of variety of economical, financial, scientific, and manufacturing data sets [10]. From the viewpoint of our research the most interesting application is the PicSOM, which is a content-based interactive image retrieval system [11]. It clusters images using separately color, texture, and shape features. A user chooses what kind

of features he would like to use and picks up a set of images that are similar to his query. The system uses SOM maps to select new images and presents them back to the user. The feature SOM maps highlight the areas on the map that correspond to the features of the set of currently selected images. The interaction continues until the user reaches his goal.

Our research described below is devoted to creating a method for unsupervised classification of events in multicamera indoors surveillance video and visualization of results. It is a part of the Multiple Sensor Indoor Surveillance project that is described in the next section. Then we describe sequentially our data collection and preprocessing procedure, one- and two-level clustering using SOM, techniques for detecting rare events, an approach to classification of new events using Gaussian mixture models (GMM) that are derived from SOM map, and a tool for visualization and browsing results. Finally, we summarize the results and speculate on the future work.

5.2 Multiple Sensor Indoor Surveillance Project

This research is a part of the Multiple Sensor Indoor Surveillance (MSIS) project. The backbone of the project consists of 32 AXIS-2100 webcams, a PTZ camera with infrared mode, a fingerprint reader, and an infrared badge ID system that has 91 readers attached to the ceiling. All this equipment is sensing an office floor for Accenture Technology Labs. The webcams and infrared badge system cover two entrances, seven laboratories and demonstration rooms, two meeting rooms, four major hallways, four open-space cube areas, two discussion areas, and an elevator waiting hall. Some areas overlap with up to four cameras. The total area covered is about 18,000 ft^2 (1,670 m^2). The fingerprint reader is installed at the entrance and allows matching an employee with his or her visual representation. The backbone architecture also includes several computers, with each computer receiving signals from 3–4 webcams, detecting "events" and recording the images for that event in JPEG format. The event is defined as any movement in the camera's field of view. The signal sampling frequency is not stable and on average is about 3 frames per second. The computer also creates an event record in an SQL database. Events detected by the infrared badge ID system and the results of face recognition using PTZ cameras go to the other database. The event databases serve as a common repository for both people who are doing manual search of events and automatic analysis.

The objectives of the MSIS project are to

- Create a realistic multisensor indoor surveillance environment;
- Create an around-the-clock working surveillance system that accumulates data in a database for three consecutive days and has a GUI for search and browsing; and
- Use this surveillance system as a base for developing more advanced event analysis algorithms, such as people recognition and tracking, using collaborating agents, and domain knowledge.

The following analyses and prototypes have been developed or are planned to be developed in the nearest future:

- Searching and browsing of the Event Repository database using a Web browser.
- Creating an event classification and clustering system.
- Counting how many people are on the floor.
- Creating a people localization system that is based on evidence from multiple sensors and domain knowledge (see Chapter 21 in this book for details).
- Creating an awareness map that shows a person's location and what he or she is doing at any given moment.
- Creating a real-time people tracking system that gives an optimal view of a person based on prediction of the person's behavior.
- Creating a system that recognizes people at a particular location and interacts with them via voice messaging.

The below-described research was inspired by the following practical problem: how automatically classify and visualize a 24-h video captured by 32 cameras?

First we implemented a Web-based tool that allows users searching and browsing the Event Repository by specifying the time interval and a set of cameras of interest. The tool's output consists of a sequence of events sorted by time or by camera and time. Each event is represented by a key frame and has links to the event's sequence of frames. Using the tool, a user can quickly sort out events for a time interval that is as short as 1–2 h, but can be overloaded with a large number of events that occurred during 24 h, which counts from 300 to 800 events per camera. The tool does not give the user a "big picture" and is useless for searching for rare events. These reasons motivated our research for unsupervised classification of events.

5.3 Data Collection and Preprocessing

Our raw data are JPEG images of size 640 by 480 pixels that are captured by AXIS-2100 webcams at the rate 2–6 Hz. Each image has a time stamp in seconds passed from the midnight of the day under consideration. To synchronize images' time stamps taken by different computers, we used an atomic clock program to set up time on each computer. The background subtraction algorithm is applied to each image to extract foreground pixels. We use two approaches for background modeling—an adaptive single frame selection and estimating median value for each pixel using a pool of recent images. After subtracting the background, morphological operations are applied to remove noise. Then the following features are extracted from the image:

- Motion features that characterize the foreground pixels' distribution (64 values). The foreground pixels' distribution is calculated on an 8-by-8 grid, and the value for each cell of the grid is the number of foreground pixels in the cell divided by the cell's area.
- Color histogram (8 bins) of the foreground pixels in the RGB color space ($3 * 8 = 24$ values).

Then the above data are integrated by tick and by event. The *tick* is a time interval which it is set up to 1 s in our case. The notion of tick and its value is important

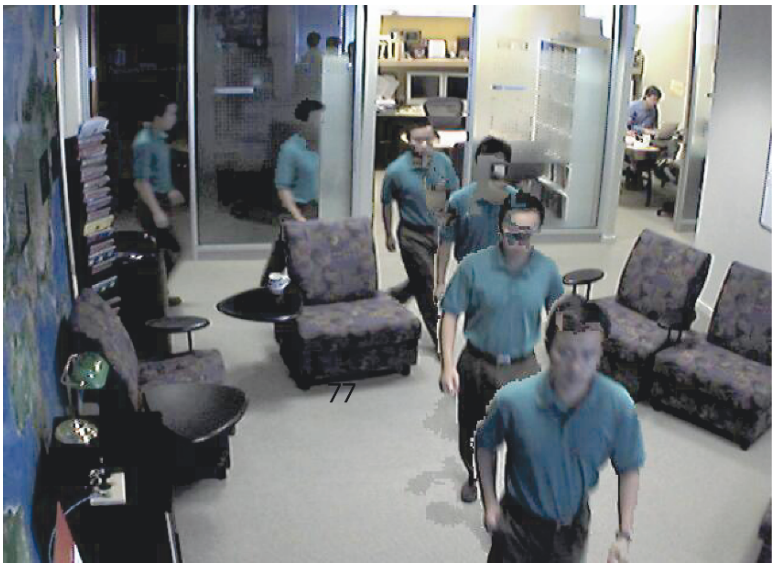

Fig. 5.1. Summary frame for an event.

because we deal with multiple nonsynchronized cameras with overlapping fields of view. Ticks allow loosely—up to the tick—synchronizing cameras' data. They also allow regularizing frame time series taken with varying sampling rates. The data integration by tick and event consists of averaging motion and color data. For visual representation of a tick or an event, a "summary" frame is created. It accumulates all foreground pixels of all images from the tick/event into one image. The summary frames serve as key frames for representing ticks/events. Figure 5.1 gives an example of a summary frame of an event.

Before presenting details of our approach for estimating an event boundaries, let us consider what kind of events we can expect to find in an indoor office environment. A camera can watch a hallway, a meeting room, a working space such as cubicles or laboratories, a recreational area such as a coffee room, or a multipurpose area that is used differently at different time of the day.

Let us assume that we are using a percentage of foreground pixels F in the image as an integral measure of motion. If the camera watches a hallway, then most events present people walking along the hallway, getting in and out of offices, or standing and talking to each other. Most events last for seconds and some for several minutes. In this case the plot of F over time looks as number of peaks that represent short events and some trapezoidal "bumps" that correspond to longer events (Figure 5.2a). In Figures 5.2 and 5.3 the x-axis presents time in seconds and y-axis presents the average percentage of foreground pixels (F-measure) during a tick. Some bumps can have peaks that correspond to combinations of transient and long-term events. If a camera watches a meeting room, then we have long periods of time when the room is empty ($F = 0$) that interchange with periods when a meeting is in process. The latter has a trapezoidal F-plot with a long base and some volatility that corresponds to

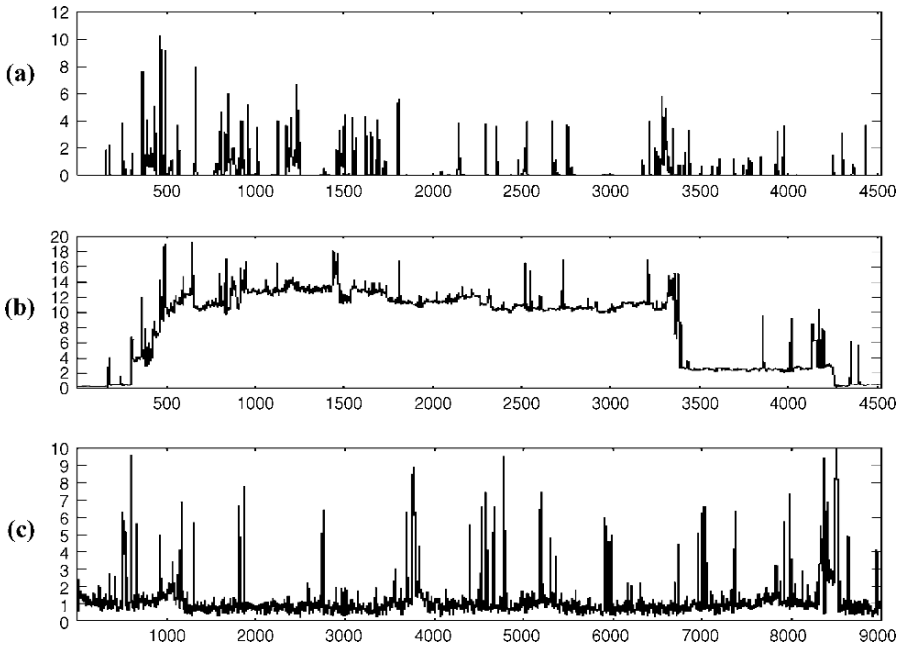

Fig. 5.2. Foreground pixels patterns for cameras that watching different locations.

people movement during the meeting and some small peaks that correspond to events when participants arriving and leaving the room (Figure 5.2b). In case of a recreational area camera, the events are typically longer than for a hallway but much shorter than for a meeting room camera. They correspond to events when people getting into the area for drinking coffee, having lunch, reading, talking on their mobile phones, or talking to each other.

If a camera watches a busy working area such as cubicles or laboratories, then we have such events as people arriving at and leaving their working places, sitting down and standing up, moving and communicating with each other. The F-plot for such camera never goes to zero at working hours and looks like a long volatile meeting (Figure 5.2c).

Having watched F-plots for several cameras, we came to the conclusion that, first, F-measure can be used for event boundaries estimation, and, second, the notion of event is a relative one. For example, on the one hand, the whole meeting can be considered as an event, but, on the other hand, it may be considered as a sequence of events, such as, people arriving for the meeting, participating in the meeting, and leaving the room. Each of these events can be divided into shorter events down to a noticeable movement. These observations encouraged us to use the wavelet decomposition of F-signal to detect events at different levels. We used the wavelet decomposition of levels from 2 to 5 with Haar wavelet for calculating approximation and details. Then we applied a threshold to approximation signal that slightly exceeds the noise level to find the boundaries of major events such as meetings (mega-events),

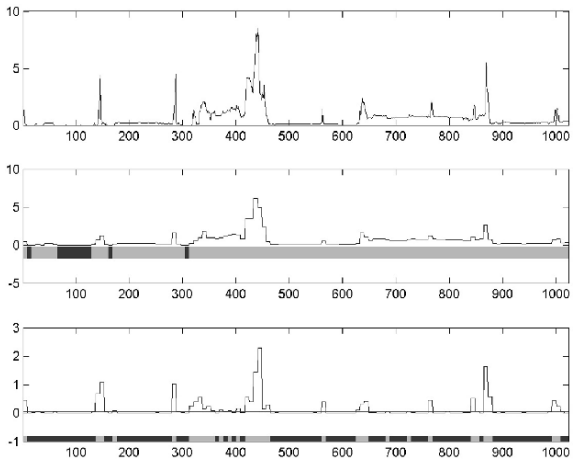

Fig. 5.3. Event boundaries detection using Haar wavelets.

and another threshold to the absolute value of highest detail signal to find internal events (micro-events). Figure 5.3 presents the results of event detection for a signal of length 1024 s using wavelet decomposition of level 3. The top plot presents the original F-signal; the middle plot is its level 3 approximation, and the bottom plot presents its level 3 details. The boundaries of macro- and micro-events presented in the lower parts of the middle and bottom plots correspondingly. The comparison of automatic extraction of micro-events boundaries with manual extraction gives a high agreement coefficient (75–90%) but people tend to detect less events, but the events are more meaningful. The boundaries of micro events are used for data integration.

After integrating data by tick and by event, we have two sets of data for each camera. The tick-level data set record consists of the following elements: tick value from the beginning of the day, names of the first and last frames of the tick, and integrated motion and color data. The event-level data set record consists of a unique event identification number, first and last tick values, names of the first and last frames of the event, and integrated motion and color data. We used both of these data sets for unsupervised classification and visualization presented below.

5.4 Unsupervised Learning Using Self-Organizing Maps

We applied the self-organizing map approach to tick/event data for clustering and visualization. We use 2D rectangular maps with hexagonal elements and Gaussian neighborhood kernels.

5.4.1 One-Level Clustering Using SOM

In one-level clustering approach we use both motion and color data to build the map. For creating maps we used the SOM Toolbox for MATLAB developed at the Helsinki

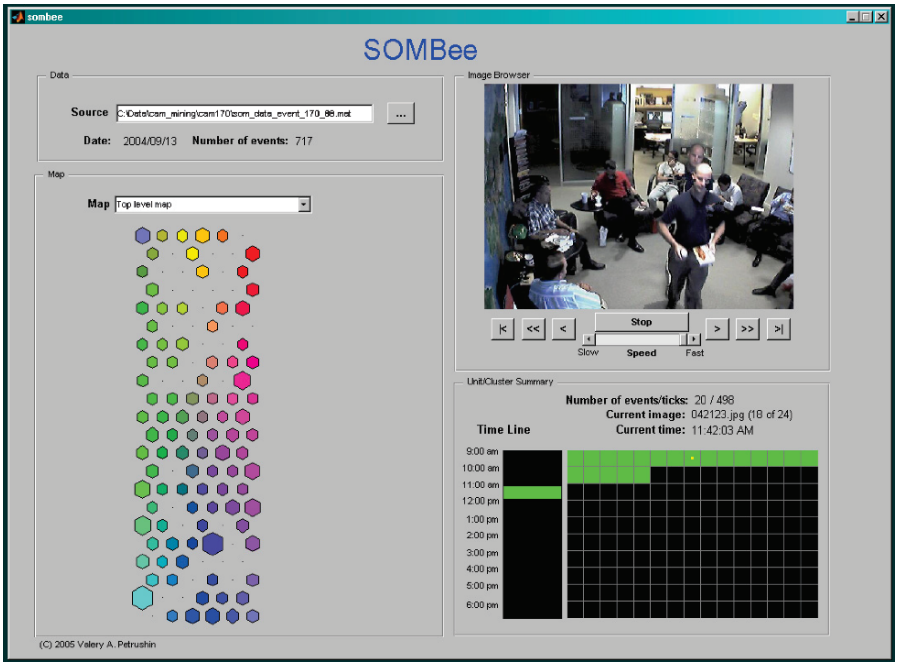

Fig. 5.4. Visualization tool displays a self-organizing map for event data.

University of Technology [12]. The toolbox allows displaying maps in many ways varying the units' sizes and colors. In our experiments we found that the size of maps for tick data can reach 1400 units (70 by 20) and for event data—about 250 units (25 by 10). Figure 5.4 presents the map for event data of a camera that observes a multipurpose area. It presents 717 events on 22-by-6 lattice (132 units). The unit's size reflects the number of events attracted by this unit (the number of hits). The unit's size is calculated using the formula (5.1).

$$s(u) = 0.5 \cdot (1 + \frac{hits(u)}{\max\limits_{u} hits(u)}) \tag{5.1}$$

where $s(u)$ is the size of unit u, and *hits(u)* is the number of data points attracted by the unit u. It means that if a unit has at least one hit then its size is not less than 0.5, and the size of the unit is proportional to the number of hits. The units with zero hits are not displayed. The units' colors show the topological similarity of the prototype vectors.

A visualization tool that is described below allows exploring the contents of each unit. However, the number of units can be large that makes unit browsing very laborious. Next step is to apply the k-means algorithm for clustering map units (prototype vectors) [13]. As the number of units is about one order smaller than the number of raw data, we can run the k-means algorithm with different number of intended clusters, sort

the results according to their Davies–Bouldin indexes [14], and allow users browsing clusters of units. The Davies–Bouldin index of a partitioning $P = (C_1, C_2, \ldots, C_L)$ is specified by the formula (5.2),

$$DBI(P) = \frac{1}{L}\sum_{i=1}^{L} \max_{i \neq j} \left\{ \frac{S(C_i) + S(C_j)}{D(C_i, C_j)} \right\} \tag{5.2}$$

where C_i, $i = \overline{1, L}$ are clusters, $S(C) = \dfrac{\sum_{k=1}^{N} \|x_k - c\|}{N}$ is the within-cluster distance of the cluster C, which has elements x_k, $k = \overline{1, N}$ and the centroid $c = \frac{1}{N}\sum_{k=1}^{N} x_k$, and $D(C_i, C_j) = \|c_i - c_j\|$ is the distance between clusters' centroids (between-cluster distance).

We do k-means clustering of units for the number of clusters from 2 to 15 for event data and from 2 to 20 for tick data. Then the visualization tool allows the user to pick up a clustering from a menu where each clustering is shown with its Davies–Bouldin index. Figure 5.5 presents the clustered map for the event data of the same camera and a legend that shows the number of events in each cluster. The centroid unit of each cluster is presented in complementary color. When the number of clusters is large, some clusters may consists of several transitional units that have no raw data (events or ticks) associated with them. The visualization tool allows browsing events or ticks by cluster.

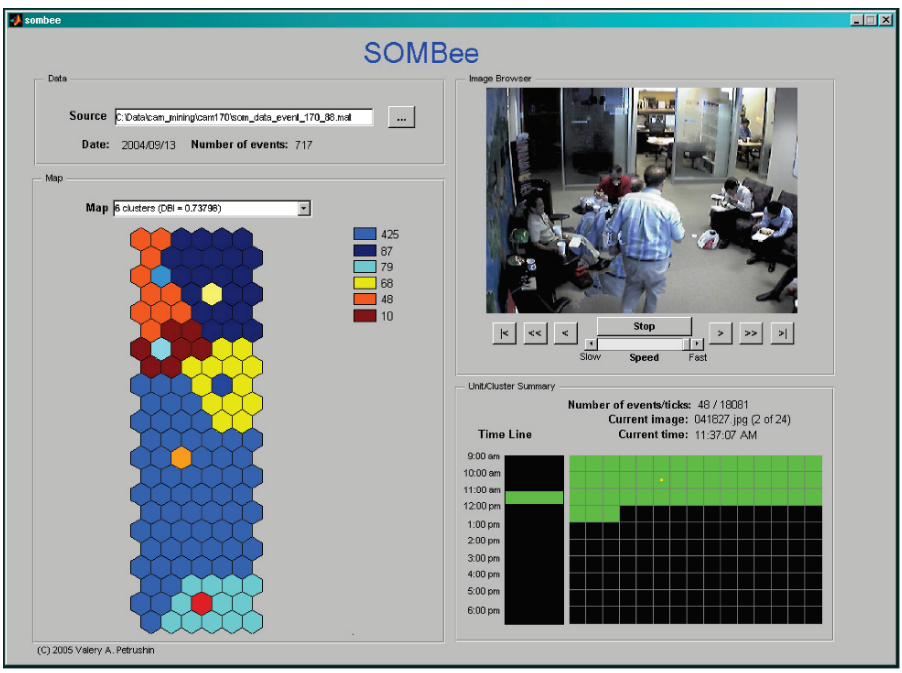

Fig. 5.5. Visualization tool displays the results of clustering of SOM units for event data.

5.4.2 Two-Level Clustering Using SOM

Our raw data have two kinds of features: motion and color features. In two-level clustering we explore these features consequentially. On the top level, only motion features are used to create the main map. Then we use k-means clustering for the map units as we described above. After this, for each obtained cluster we build a SOM map using color features. Such separation of features allows differentiating more precisely spatial events and easier detecting unusual events. In indoor environment, where most of moving objects are people, who change their clothes every day, the variance of color features is higher than the variance of motion features. Separating motion features allows collecting them over longer periods and creating more robust classifiers. To create a classifier, we accumulated motion data during a week, built an SOM map, and applied k-means clustering to its units. The whole SOM map M can be considered as a Gaussian Mixture Model (GMM) [15] with the probability density function represented by (5.3).

$$f(x|M) = \sum_{i=1}^{N} w_i \cdot f_i(x \mid \lambda_i) \text{ and } \sum_{i=1}^{N} w_i = 1 \tag{5.3}$$

where $f(\cdot \mid \lambda_k)$is the probability density function for model λ_k, $\lambda_k = \mathcal{N}(\mu_k, \Sigma_k)$ is a Gaussian model for k-th unit with mean μ_k and covariance matrix Σ_k, N is the number of units in the map. It is often assumed that the covariance matrix is diagonal.

Each unit of the map has a Gaussian kernel associated with it and a weight w_i, which is proportional to the number of data points attracted by the unit (the number of hits for this unit). The log-likelihood of a data point x to belongs to the map is estimated using Equation (5.4).

$$\log\ L(x \mid M) = \log\left(\sum_{i=1}^{N} w_i \cdot f(x \mid \lambda_i)\right) \tag{5.4}$$

On the other hand, for a partitioning $P = (C_1, C_2, \ldots, C_L)$ each cluster can be viewed as a GMM with the log-likelihood function represented by (5.5).

$$\log\ L(x \mid C_k) = \log\left(\sum_{i \in C_k} w_i \cdot f(x|\lambda_i)\right) \tag{5.5}$$

A new piece of data can be classified by calculating likelihood using GMM associated with each cluster and assigning the new data to the cluster with maximal likelihood (5.6). The same procedure can be applied to color features. Combining motion-based classifiers on top level with color-based classifiers on the second level, we obtain a hierarchical classifier.

$$C_{k^*} = \arg\max_{C_k} \{\log L(x|C_k)\} \tag{5.6}$$

5.4.3 Finding Unusual Events

In some cases, finding unusual events is of special interest. But what we should count as an unusual event is often uncertain and requires additional consideration. An event can be unusual because it happened at unusual time or at unusual place or had unusual appearance. For example, finding a person working in his office at midnight is an unusual event, but the same event happened at noon is not. A person standing on a table would be considered as an unusual event in most office environments. Many people wearing clothes of the same color would be considered as an unusual event unless everybody is wearing a uniform. Everybody agrees that an unusual event is a rare event at given time and space point. But a rare event may not be an unusual one. For example, a person sitting in his office on weekend could be a rare but not surprising event.

After thoughtful deliberation, we decided to use computers for finding rare and frequent events leaving humans to decide how unusual or usual they are. In our research we distinguish between *local* rare/frequent events—these are events that happened during one day—and *global* rare/frequent events—those that happened during longer period of time and the surveillance system accumulated data about these events. We also distinguish between events that happened during regular working hours and out of them.

For finding local rare events, we are using an automatic procedure that indicates areas of the SOM map that contain potential rare events. This procedure assigns 1 of 13 labels for each unit of SOM map based on the number of hits attracted by the unit and the distances from the unit to its neighbors using Equation (5.7),

$$R_{nm} = \{u : \text{hits}(u) \leq H_n \cap \min_{v \in Nb(u)} \{D(u, v)\} \geq D_m\} \tag{5.7}$$

where $H_n = \{5, 10, 20, 40, ^*\}$is the list of hit levels (* stands for "any"), $D_m = \{0.9, 0.75, 0.5\} \cdot \max_{u,v \in M}(u, v)$ is list of distance levels, *Nb(u)* is a set of neighbors of the unit u, and *hits(u)* is the number of hits of unit u.

For visualizing local rare events, we use two approaches. The first is to display the SOM map with a particular color assigned to each R_{nm} class. The color spans over different shades of red–orange–yellow, depending on the n–m combination. The map allows the user identify and explore areas that have high potential to have rare events. The second approach is a 3D surface that shows distances between units of the SOM map and indicates how many data points (hits) belong to each unit using markers with sizes that are proportional to the number of hits. Figure 5.6 shows the 3D visualization for event data. A user can rotate the axes searching manually for "highlanders"—small unit markers that are located on the top of peaks or for "isolated villages" which are sets of small unit markers that are located in closed "mountain valleys". Using the combination of (semi-)automatic and manual approaches allows the user finding promptly local rare events.

For detecting global rare events, the following procedure is proposed. First, the GMM classifier is applied to a new event/tick motion data. If it gives a high probability for a small cluster, then the system declares that a rare event of particular type is found.

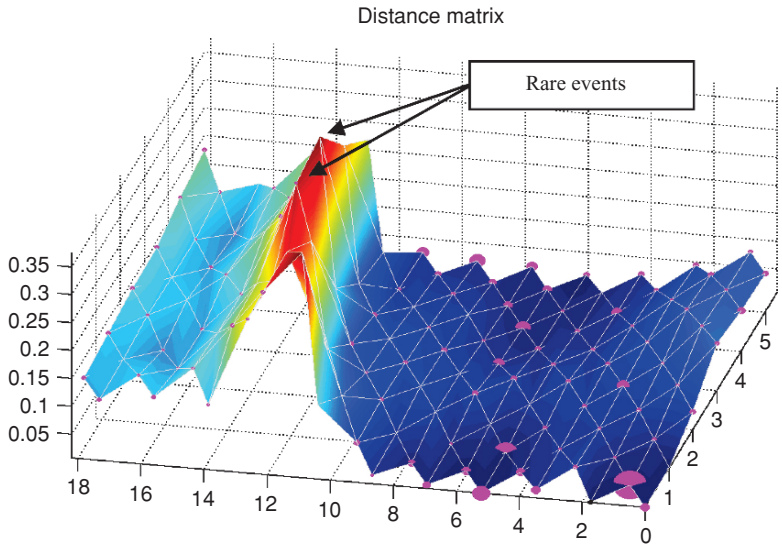

Fig. 5.6. Visualization of distances between units for searching for rare events.

If the classifier gets low probabilities for all clusters then the system indicates that it is a new (and rare) spatial event. Such events are accumulated and can be used for building a new version of GMM classifier. In case when the event belongs to a moderate or frequent event cluster, the system applies the corresponding color-based GMM classifier to detect rare or new events regarding their color features.

5.5 Visualization Tool

The above-described techniques have been integrated into an event visualization tool. Figures 5.4 and 5.5 show snapshots of the tool. The tool's GUI consists of four panels—Data, Map, Image Browsers, and Unit/Cluster Summary. Using the Data panel, the user selects and loads data. Currently each camera has a separate file for its tick and event data. The file contains preprocessed raw data and SOM data for chosen SOM architecture—one- or two-level SOM map. The user selects the desired map from a menu. The map is displayed in the Map panel or in a separate window.

The Map panel displays the current SOM map, which could be the top-level map that shows color coded units with unit sizes reflecting the number of hits, a cluster map of different number of clusters with color-coded units, or a map for indicating potential rare events. The 3D surface that presents distances between units and the number of hits in each unit is displayed in a separate window (see Figure 5.6). The user can rotate the 3D surface for hunting for local rare events.

When the user clicks on a unit or a cluster of units on the SOM map, the contents of the unit or cluster is displayed in the Unit/Cluster Summary panel. This panel presents information about the current item (unit or cluster). It shows the number of events

and/or ticks in the current item, the name of image displayed in the Image Browser panel and its time. It also has two plots. The left bar plot shows the distribution of event or tick data in time. The right plot shows all data (ticks or events) that are related to the current item. A small square corresponds to each piece of data. The color of the square indicates the time interval that the piece of data belongs to. When the user clicks on a square, the corresponding summary frame is displayed in the Image Browser panel.

The Image Browser panel displays the visual information related to the current selected tick or event in the Unit/Cluster Summary panel. Using the browser's control buttons, the user can watch the first or last frame of the tick/event, go through the tick/event frame by frame forward and backward, and watch a slide show going in both directions. The speed of the slide show is controlled by the speed slider. Clicking on the image brings the current frame in full size into a separate window allowing the user to see details. This feature proved to be very useful for exploring busy summary images.

5.6 Summary

We described an approach to unsupervised classification and visualization of surveillance data captured by multiple cameras. The approach is based on self-organizing maps and enables us to efficiently search for rare and frequent events. It also allows us creating robust classifiers to identify incoming events in real time. A pilot experiment with several volunteers who used the visualization tool for browsing events and searching for rare events showed both its high efficiency and positive feedback about its GUI.

Although we applied this approach for indoor surveillance in office environment, we believe that it is applicable in the larger context of creating robust and scalable systems that can classify and visualize data for any surveillance environment. The real bottleneck of the approach is not creating SOM maps (it takes just several minutes to create a map for 24-h tick data with 86400 records), but feature extraction and data aggregation.

In the future we plan to extend our approach to visualize data of a set of cameras with overlapping fields of view, embed the GMM-based classifier into visualization tool for detecting global rare events, and improve the graphical user interface.

References

1. Kanade T, Collins RT, Lipton AJ. Advances in Cooperative Multi-Sensor Video Surveillance. In: *Proc. DARPA Image Understanding Workshop*, Morgan Kaufmann, November 1998; pp. 3–24.
2. Siebel NT, Maybank S. Fusion of Multiple Tracking Algorithms for Robust People Tracking. In: *Proc. 7th European Conference on Computer Vision (ECCV 2002)*, Copenhagen, Denmark, May 2002; Vol. IV, pp. 373–387.

3. Krumm J, Harris S, Meyers B, Brumitt B, Hale M, Shafer S. Multi-camera Multi-person Tracking for EasyLiving. In: *Proc. 3rd IEEE International Workshop on Visual Surveillance*, Dublin, Ireland, July 1, 2000.
4. Ayers D, Shah M. Monitoring Human Behavior from Video taken in an Office Environment. Image and Vision Computing, October 1, 2001;19(12):833–846.
5. Oh J-H, Lee J-K, Kote S, Bandi B. Multimedia Data Mining Framework for Raw Video Sequences. In: Zaiane OR, Simoff SJ, Djeraba Ch. (Eds.) *Mining Multimedia and Complex Data*. Lecture Notes in Artificial Intelligence. Springer, 2003, Vol. 2797, pp. 18–35.
6. Zhong H, Shi J. Finding (un)usual events in video. *Tech. Report* CMU-RI-TR-03-05, 2003.
7. Amir, A., Srinivasan, S., and Ponceleon, D. Efficient video browsing using multiple synchronized views. In: A. Rosenfeld, D. Doermann, and D. DeMenthon (Eds.) *Video Mining*, Kluwer Academic Publishers, 2003, Vol. 1–30.
8. Gong Y. Audio and visual content summarization of a video program. In: Furht B, Marques O. (Eds.) *Handbook of Video Databases*. Design and Applications, CRC Press, 2004; 245–277.
9. Kohonen T. *Self-Organizing Maps*. Springer-Verlag, 1997.
10. Oja E, Kaski S. (Eds.) *Kohonen Maps*. Elsevier, 1999.
11. Laaksonen JT, Koskela JM, Laakso SP, Oja E. PicSOM—content-based image retrieval with self-organizing maps. *Pattern Recognition Letters*, 2000;21(13/14):1199–1207.
12. Vesanto J, Himberg J, Alhoniemi E, Parhankangas J. *SOM Toolbox for Matlab 5*, Helsinki University of Technology, Report A57, 2000.
13. Vesanto J, Alhoniemi I. Clustering of self-organizing maps. *IEEE Trans. on Neural Network*, 2000;11(3):586–600.
14. Davies DL, Bouldin DW. A cluster separation measure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1979; PAMI-1:224–227.
15. McLachlan G, Peel D. *Finite Mixture Models*, Wiley, 2000.